

[To Ted Nelson Home Page](#)

ZigZag™ Tech Notes, 00.03.26

NO ICONS • NO METAPHORS • NO FILES • NO APPLICATIONS

It's all about reference structure. What can you talk about clearly?

Note: "ZigZag" is a trademark of Project Xanadu.

ZIGZAG STRUCTURE [for those who love technicality]

There are many paradigms of computer, program and data structure. The usual ones are sequence, hierarchy, tables and OO (arguably a combination of hierarchy and tables). People have been effectively ordered to fit everything into these structures, but of course a lot of things fit badly. The real problem is to represent *true structure*: that is, the relations we see, exactly as we see them-- especially overlap, interpenetration, criss-crossing interconnection, parallel documents, disagreement and deep annotation. (Not to mention the problem of coordinating the ever-growing mass of content across many machines and storage representations.)

ZigZag structure (ZZstructure for short) allows us to represent and refer conveniently to all these things, and perhaps to coordinate them on a grand scale. It presents its own layers of overhead, but so does everything worthwhile (eg TCP/IP). It does not refer to anything in conventional ways (meaning "files").

ZZstructure consists of multidimensional lists viewable any way you like (typically in two or more orthogonal dimensions). It may be thought of as spreadsheet without the unnecessary cells, without the rectangular constraint, with as many dimensions as you like, and with looping allowed. It has also been called "the spreadsheet as redesigned by Bugs Bunny".

It could also be called a highly principled reference structure with crystalline properties (repeated sets of elements with replicated sets of connections at different angles).

ZZstructure is a simple relational quark which can represent any other structure, (including sequence, hierarchy, tables and OO) by interpretation. You can build it into anything you like.

One definition:

Elements connected by pairwise symmetrical untyped links--

- having a nominal ordering where $a > b$ or $b > a$.

- [special exceptional case: loop of two elements where both $a > b$ and $b > a$]

An element may have such links

- in as many dimensions as you like (new dimensions may be created on the fly)
- with whatever visualizations you want (crisscross visualization of any two dimensions supplied in the prototype).

THE TWO IMPLEMENTATIONS

There are two implementations, by two fabulous and brilliant guys.

1. The prototype

The prototype is in Perl, by Andrew Pam of Xanadu Australia (xanni@xanadu.net), downloadable via Xanadu.com.

It is currently at version 0.70.

The prototype comes in two flavors:

- BOOT FLOPPY (Much less trouble for most people) for stock PC (486+). This actually contains a small version of Linux and is intended not to touch your hard drive.
- Mountable on your Linux system.

DIRECTIONS FOR USE OF PROTOTYPE are at <http://www.xanadu.net/zigzag/zzDirexCondensed.html>

However, the prototype is now in maintenance mode. Active development is--

2. "Gzigzag"--

Gzigzag is the Java version, at sourceforge.net, under direction of Tuomas J. Lukka, University of Jyväskylä (lukka@iki.fi). Experienced programmers can use the CVS version manager to download the latest version.

This is in principle Open Source (LGPL and Xanadu transpublishing licenses). However, the full specs are still being hashed out and so we are not quite prepared to harness programmers for full-bore Linux-style smurf-brigade development.

DIFFERENCES BETWEEN THE TWO SYSTEMS

For practice and for developing data sets, the prototype is still the one to use. It is also EASY FOR NON-PROGRAMMERS.

Therefore the prototype may get another year or so of use.

Gzz, the Java version, is the cutting edge, with some exciting new visualizations. HOWEVER, it is much harder to use: the rather well-tuned (ahem) interface of the prototype has not yet been duplicated.

Unfortunately right now the two are completely different and incompatible. However, we hope to make it possible for Gzz to accept files developed on the prototype.

INSTALLATION OF GZZ ON PCs

Right now you have to use CVS to install Gzz. However, in principle a directory or an executable can be downloaded easily and mounted on any Java runtime environment. (I've had it working nicely on the PC.)

Installation note: Autoexec.bat should have both a correct path through the Java environment and a correct classpath. These can be typed in each time you boot up, but do you want to?

Assuming a correct installation, you must currently invoke Gzz from MS-DOS as follows:

```
java foo.Main filename
```

The file contains your entire data system, including the two current views and cursor positions.

Fresh files (sometimes unwanted)

Note that if you don't name an existing desired filename correctly, it will fire up a new one for you, which you have to notice is not what you wanted. This can be annoying, and there should be a notification soon telling you that it's firing up a virgin copy, so you don't go through the bootup unnecessarily.

Backing up:

It is easy to back up dynamically. You may copy the current file directly out of your directory *while Gzz is running*, preserving state as of that instant-- rather like lifting an egg out from under a hen.

USER FUNCTIONS PRESENTLY WORKING IN GZZ

Main cursor operations--

Move left cursor: typist's star pattern

- e s f c

up, left, right, down

- d

forward (negward) on z-axis

- D

backward (posward) on z-axis

Move right cursor: typist's star pattern

- i j l[small L] ,[comma]

up, left, right, down

- k

forward (negward) on z-axis

- K

backward (posward) on z-axis

Rotation of views:

- x

reassign x-axis right window

- X

reassign x-axis left window

- Correspondingly for left and right y and z axes.

Editing:

Use TAB to get in and out of editing mode. (Unfortunately this is much too easy to do.

OTHER FUNCTIONS

Other functions, below, describe the version of approximately 00.03.15, and are not guaranteed to stay the same.

All the available functions may be found on the "Actions" corner-list, downward from Home (+d.2).

"Bindings" may be found on the "Bindings" corner-list, +d.2 from Home.

You may easily reassign the keys which activate these functions by editing the lefthand cell.

CONNECTIONS, DISCONNECTIONS

Unfortunately connections and disconnections are handled only by absolute dimension reference, and are very hard to handle.

To connect:

Mark the successive cells to be connected (move out of them in d.1 or d.2 while shifting-- the same moves unmark). Then one of the CNCT functions (not CNCTI or CNCTC) will connect all of the marked cells in the specified direction of the current right-hand view.

BEWARE: This connective function is destructive, breaking any connections inconsistent with what you asked for. It is extremely easy to get an undesired result, so you should definitely back up before each connection operation, perhaps even before each disconnection operation.

WHERE THIS IS LEADING: THE FLOATING WORLD™ DESIGN

Floating World (also to be Open Source) is my design for an alternative computer universe. The fundamental unit is the Hyperflob, or multidimensional flying object; hyperflobs may have any number of coordinates. (The plan is to build on OpenGL graphics software optimized for stock PC graphics boards.)

Since ZZstructure easily supports any number of coordinates, multiple dimensions are easy. Hyperflobs will be defined in 3 1/2D for easy interpenetration.

The FloatingWorld spec, in rough draft since 99.11, will either be updated or published shortly in its present form.

Floating World is intended as a software universe with a new core, having all functions available under the hood in customizable ZZstructure. We expect the creation of new functions and programs (applitudes) to be much easier than in any previous environment.

FLOATING WORLD AND XANADU(R) SOFTWARE

For text and other fluid media (having separable countable elements), Floating World will use the Xanadu system of interconnection (see xanadu.com for details). It is intended as a compatible client for the xu88 system (formerly Xanadu 88.1, now available Open Source at udanax.com). All text, audio and video will be subject to Xanadu-type content links and transclusion using address comparison.

Anyone interested in the Udanax software should contact Roger Gregory (roger@xanadu.net), who has done a magnificent job in keeping this work alive.